Anonymization of the University Information System Log Data: a Case Study

Jiri Zettel

Department of Information and Knowledge Engineering, University of Economics, Prague
W. Churchill Sq. 4, 130 67 Prague, Czech Republic
xzetj01@vse.cz

Abstract. This paper brings deep insight into data preparation when implementing group-based anonymization techniques. A real-world dataset contains access log data and is used for the consequent anomaly detection task. Unlike other research in the field of anonymization, we don't focus on the design of new algorithms, but on the pre-processing steps and on exploring of applicability of existing algorithms. Each algorithm has specific requirements for the data, so pre-processing must be comprehensive. In this paper we present how such data can be transformed into relational data, introduce a novel approach for anonymization of IPv4 address in our dataset using several anonymization algorithms and discuss their principles, strengths, and weaknesses. Two ways of pre-processing of IPv4 for k-anonymity algorithms are presented: first, we split IPv4 into four parts and create generalization hierarchies and second we convert IPv4 to integer values. We propose an improvement in Mondrian algorithm suitable for categorical attributes which gives better results than the original algorithm.

Keywords: Anonymization, K-Anonymity, IPv4, Privacy-Preserving, Anomaly Detection, Data Preparation

1 Introduction

There is an ongoing university research project with the objectives (i) to detect computer attacks in the university information system using anomaly-based detection methods and (ii) to create an artificial generator of such attacks. The dataset used for experiments contains access log data and is represented in a relational database. The results of the experiments on the datasets should be published, therefore the dataset has to be anonymized. Anonymization is a technique of changing data in a way that prevents the identification of a person. There is a tradeoff between data utility and a level of anonymization. Our goal is to experiment with anonymization techniques so that the risk of re-identification of a person stays at an acceptable level and at the same time the information required for successful anomaly detection remains in the dataset. In this paper, we selected and evaluate group-based anonymizations. A review of such techniques is presented in [1]. Consider that e.g. IP address is anonymized in a way that last two octets are removed. If the dataset contains the only IP address starting 146.102.*.* then this anonymization technique does not protect the person using this address. The same logic applies when IP is replaced by a unique identifier, the pattern of a particular person can be tracked and identity can be compromised if an attacker knows any additional information which is the usual case. With the k-anonymous group, each anonymized IP address belongs to a group with k-1 other IPs. The main contribution of this paper contains (i) the data preparation steps for various anonymization algorithms, including transformation to relational data (ii) their comparison, evaluation and possible optimization and (iii) a novel approach for anonymization of the dataset with IPv4 addresses. Two ways of pre-processing convenient for the use by k-anonymity algorithms are presented further, we also discuss the ideas behind each decision. The experiments with Mondrian led us to optimize it for the use of categorical attributes.

2 Background and Related Work

Most of the research focuses on discovering models guaranteeing privacy and designing new algorithms on how to achieve it. Such benchmarks utilize the same public dataset *Adult*[1] to prove how the new algorithms outperform the others. Main established models still in use are k-anonymity [2], ℓ-diversity [3], t-closeness [4], ε-differential privacy [5] or ρ-uncertainty [6]. An overview of the data anonymization methods was described by Prasser [7], author of ARX anonymization tool. Studies dealing with the application of anonymization algorithm on real-world data are rare. One of such case studies describes the anonymization of medical surveys [8] using k-anonymity. Emam described a framework for anonymization of clinical data [9]. Ayala-Rivera presented a systematic evaluation of k-anonymization algorithms on the *Adult* dataset. Differential privacy technique is currently being adopted in the commercial sector [10]. For the anonymization of IPv4 address, according to the survey of network traffic anonymization methods [11], common methods are prefix-preserving, random permutation, truncation or grouping. None of the reviewed IPv4 papers deal with k-anonymization algorithms.

K-Anonymity ensures that each tuple in a table is indistinguishable from at least *k* others, with respect to quasi-identifiers *(QI)*. *QI* are attributes whose release must be controlled. Achieving k-anonymity is through searching the minimal *generalization* of the values of the attributes and optionally through tuples *suppression*. The relationship of the domain levels forms the *domain generalization hierarchy (DGH)*. A *value generalization hierarchy (VGH)* represents a relationship between the values in the domains. For the evaluation of the anonymization, we use the Anonymization ToolBox[2]. The algorithm *Datafly* [12] uses a greedy heuristic, *Incognito* [13] performs hierarchy-based optimal search and *Mondrian* [14] performs partitioning. The first metric we will use in the evaluation is the discernibility metric, it assigns a penalty to each tuple based on how many tuples in the transformed dataset are indistinguishable from it [15]. The second metric is the normalized average equivalence class size metric described in [14].

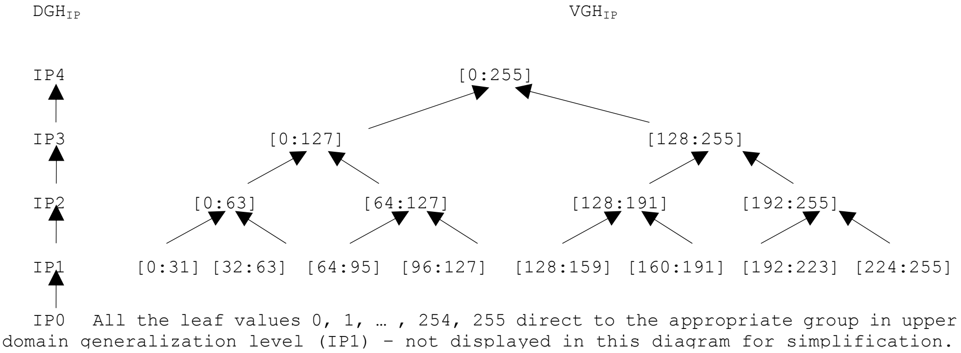
[1] Data set can be found in UCI Repository at: https://archive.ics.uci.edu/ml/datasets

[2] UTD Anonymization Toolbox available at: http://cs.utdallas.edu/dspl/cgi-bin/toolbox/

3 Data Understanding and Preparation

The data used in this experiment are represented by HTTP requests stored in a MySQL table, one per user action. User-supplied POST parameters we removed because they are sensitive. *UserID* and *SourceIP* are quasi-identifiers. They can lead to the identification of the persons when linking to some additional data. Another quasi-identifier could be a timestamp. There are some indications that the identity could be revealed when discovering user behavior. We selected a subset containing 350k transactions, activities generated in one day. Some modifications in the data were necessary to be done first. For example, *SourceIP* was transformed into five new attributes, first having integer values (using INET_ATON function[3]) and remaining four having IP address split into four octets. Then we extracted the identities from the transactional data and thus created the relational data with all the users and their originating IPs. This new table consists only *UserID* and *SourceIP* attributes and the relationship between the attributes is many-to-many. Statistics are described in Table 1.

Full-domain Generalizations. Datafly and Incognito work with categorical attributes. All the values within the domain have to be generalized to the same level in DGH. This will lead to over-generalization if there are values which vary greatly from all the other values in the domain because they need to be generalized more to fit in an equivalence class. And this is the case for the IP addresses. IP split into four parts will represent the IP as four separate quasi-identifiers and allow to apply different domain generalization level to each IP part. In fact, it's easier to generalize the fourth octet than the first one because its values are distributed more evenly. The last octet represents the host in the subnet, the first octet is assigned by IANA[4] and the next one by Regional Internet Registries. VGH/DGH created are illustrated in Fig. 1.



DGH_IP
VGH_IP
IP4
[0:255]
IP3
[0:127]
[128:255]
IP2
[0:63]
[64:127]
[128:191]
[192:255]
IP1
[0:31]
[32:63]
[64:95]
[96:127]
[128:159]
[160:191]
[192:223]
[224:255]
IP0
All the leaf values 0, 1, … , 254, 255 direct to the appropriate group in upper domain generalization level (IP1) – not displayed in this diagram for simplification.

Fig. 1: Domain and value generalization hierarchies for IP address

[3] https://dev.mysql.com/doc/refman/8.0/en/miscellaneous-functions.html#function_inet-aton

[4] IANA IPv4 Address Space Registry available at https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

Top-down Algorithm. Mondrian partitions the values until k-anonymity is achieved so it works well with numeric attributes. *UserID* attribute is numerical discrete and IP address can be numerized also to discrete values. The advantage of numerizing IP over using VGH for partitioning is obvious. It will allow much more fine-grained generalization. We proposed an improvement in the Mondrian, which is suitable for the use of categorical attributes mapped to discrete numerical values, especially when the attributes have low cardinality. We implemented it in UTD Toolbox and further, we prove it gives better results also for the numerized IP address. The original algorithm creates a frequency set in the selected dimension and searches for median (*splitVal*). It splits the values to the left-hand side (*lhs*) and right-hand side (*rhs*) interval. *Lhs* interval is then created using inclusive interval (including *splitVal*) and *rhs* exclusive. This cut is then recursively repeated in *rhs* and *lhs* until at least *k* values are present in each interval. Table 2 shows an example frequency set, were median is 3. Creating *lhs* = [1:3] and *rhs* = (3:4] would not be allowable cut considering *k* = 10. In such cases creating intervals where *splitVal* is *rhs* inclusive would still allow further cut, so we extended the algorithm by one more step which tries to cut the partition to *rhs* inclusive when *lhs* inclusive is not allowable.

Table 1: Identities table statistics

Attribute	Distinct Count	Min	Max	Stdev	Avg
Octet 1	144	1	223	51.17	109.09
Octet 2	253	0	255	64.50	124.21
Octet 3	256	0	255	75.68	125.24
Octet 4	256	0	255	75.40	119.79
UserID	9974	-	-	-	-
Num. IP	12736	-	-	-	-

Table 2: Example frequency set

QID value	count
1	21
2	370
3	4214
4	5

4 Experimental Evaluation

Following configuration was used for the experiment: Intel Xeon CPU 1.9 GHz, Java 1.8U211 32-bit runtime-environment, maximum heap size is about 1.6GB.

Datafly for IP Address. First evaluation is done for parameters *k* = 10 and suppression threshold = 10. The time processed was 642s. Octet 1 is generalized to DGH level = IP3, octets 2 to 4 are generalized to IP2 level. The result gives 9 suppressed tuples (described as an equivalence group of size 9) and a total of 128 equivalence classes. The smallest equivalence size is 11 and the largest is 432. The distribution of the group sizes is described in Fig. 2. This shows how many distinct users are associated with each anonymized IP address. Normalized average group size is calculated to be 11.27 and discernibility metric is 2,893,151.

The second evaluation is done for *k* = 10 and suppression threshold 0. The time was 627s. To keep the suppression on zero level, generalizations are greater, octets 1 and 2 are generalized do DGH = IP3 level, while octets 3 and 4 to IP2 level. There are 64 groups, less than in the first experiment, but they are bigger (37 to 468). Normalized average group size is also higher (22.55) and discernibility metric is 4,243,430. The distribution of the group sizes is described in Fig. 3.

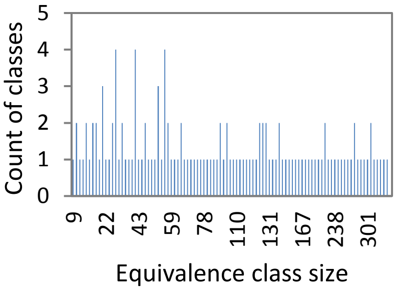


Fig. 2: Datafly, k=10, supp=10

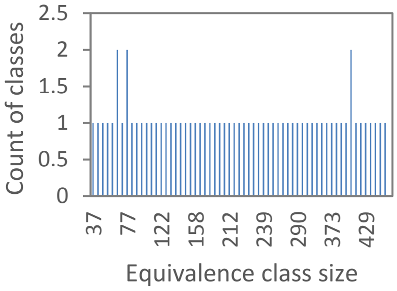


Fig. 3: Datafly, k=10, supp=0

Incognito for IP Address. Because of time limitation, we removed the IP1 level of DGH for IP address. The anonymization took 20942s which is nearly 6 hours. The optimal level of anonymization found is 3-2-3-0 for the IP parts 1 to 4, meaning the last octet remained with the original values. Intuitively the IP address should be generalized in such way, that most generalized bits should be on the right side. But this would not represent optimal anonymization by Incognito of the quasi attributes as we selected them. The solution for this would be to choose only three or two last octets. Fig. 4 shows the distribution of equivalence classes (there are total 512 of them). Normalized average group size is smaller compared to Datafly (2.82) and discernibility metric is 447,140.

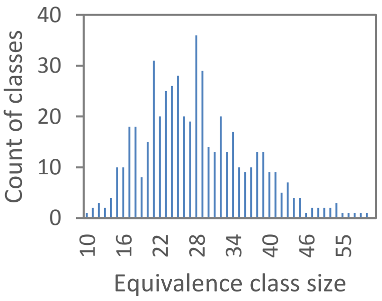


Fig. 4: Incognito, k=10, supp=10

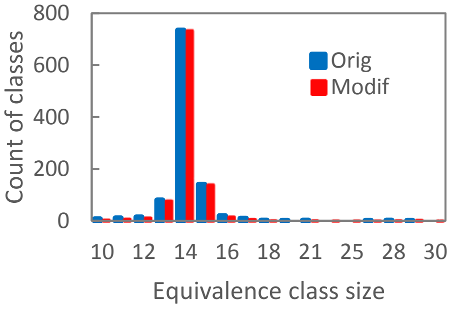


Fig. 5: Mondrian, k=10, alg. comparison

Mondrian for IP Address. First Mondrian evaluation for k = 10, using original Mondrian partitioning algorithm took 81s. There were 1016 equivalence classes created, smallest having size of 10 and the largest 30. Normalized average group size is calculated to be 1.42 and discernibility metric is 207,524. Second evaluation for the same parameter was done with the modified algorithm. Time processing was 80s, 1022 classes created in total (from size 10 to 29). Normalized average group size is calculated to be 1.41 and discernibility metric is 205,304. The results are slightly better than with the original algorithm. The graph comparison can be found in Fig. 5. To compare how many equivalence classes were created by the modified algorithm we searched for partitions split into intervals *(a:splitVal)* and *[splitVal:b)*, which are those where *splitVal* is exclusive in *lhs* and inclusive in *rhs*. There are 6 partitions in total, cut in 12 such intervals, meaning 12 equivalence classes out of 1022 were created in the case when standard algorithm did not find further allowable cut. We examined the IP addresses defining the intervals and all of them belong to Universities or Internet Service Providers. This confirms the assumption about low cardinality attributes.

Mondrian for User ID and IP Address. In this experiment, we included also anonymization of user ID within the equivalence class, along with the IP address. This will ensure that not only the IP address is indistinguishable from k – 1 other addresses, but so does the user ID. The processing time is 124s. There are 9 equivalence classes created, smallest has 10 equivalent members and largest 18. There are only 2 classes of 18 members. More information is shown in Fig. 6. Normalized average group size is calculated to be 1.37 which is much smaller than for the previous algorithms. The discernibility metric is also much smaller (199,530). The distribution of user IDs can be found in Fig. 7. Total intervals of IP addresses are 648, the distribution is shown in Fig. 8. Fig. 9 illustrates the relationship between the anonymized IP address groups and users. It shows that each IP address interval is associated with at least 10 users. Highly used IP intervals belong to the subnets of the University or the largest Internet Service Providers.

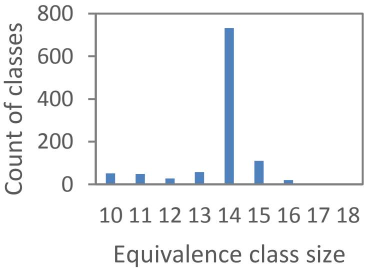


Fig. 6: Mondrian for UserID and IP, k=10

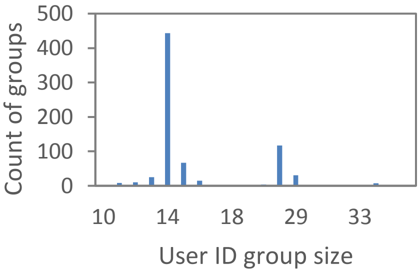
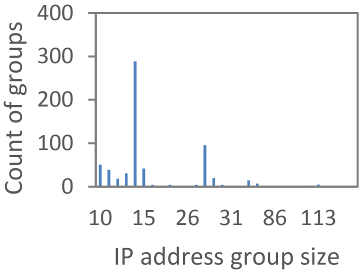


Fig. 7: Mondrian for UserID and IP, k=10



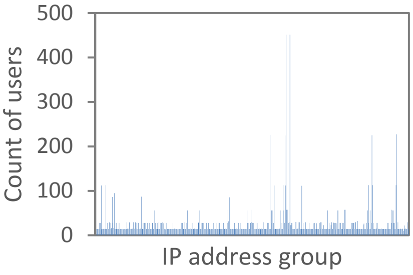


Fig. 8: Mondrian for UserID and IP, k=10

Fig. 9: Mondrian for UserID and IP, k=10

5 Conclusions and Future Work

Our work brings additional and practical information to the papers evaluating the algorithms on public datasets. We believe that it can be useful for our further experiments and can give other researchers insight into the pre-processing of raw data. The experiments proved that existing k-anonymization algorithms can be applied when data is prepared in a convenient way. The example illustrated anonymization of the IPv4 address which considers their occurrences in the dataset to be published, as opposed to the IP anonymizations that don't consider other instances in the dataset. Best results were achieved by the Mondrian algorithm because the partitioning technique is very good for the numeric continuous attributes or categorical mapped to discrete numerical values when the cardinality is high. When the cardinality is low we would consider using Datafly or Incognito with value generalization hierarchies. However, we saw the limitation of the Incognito's optimal algorithm which is computationally very intensive. The intervals created by Mondrian as equivalence groups for IP address can be easily converted back to one anonymized IP address in IPv4 format when replacing the digits differentiating on the same index in lower and upper bound by an asterisk (*) while keeping the digits with same values on the same index. We also verified that the idea of the modified Mondrian algorithm is correct in the experiment with IP addresses. In our future work, we would like to perform a similar experiment with the transaction data, mainly to anonymize the timestamp attribute to hide the user behavior pattern. Eventually, consequent experiments with the anomaly detection task on anonymized dataset need to be done to evaluate when the detection is successful.

Acknowledgment

The work reported in this paper is carried out with the support of the IGA F4/12/2019 project of the University of Economics, Prague.

References

[1] J. Zettel and P. Berka, “STUDY OF ANONYMIZATION TECHNIQUES FOR LOGGING DATA FROM UNIVERSITY INFORMATION SYSTEM,” presented at the 26th Interdisciplinary Information Management Talks IDIMT 2019, 2019.

[2] L. Sweeney, “k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, Oct. 2002.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “ℓ-Diversity: Privacy Beyond k-Anonymity,” p. 12, 2007.

[4] N. Li, T. Li, S. Venkatasubramanian, and T. Labs, “t-Closeness: Privacy Beyond k-Anonymity and -Diversity,” p. 10, 2007.

[5] C. Dwork, “Differential Privacy,” in *Automata, Languages and Programming*, 2006, pp. 1–12.

[6] J. Cao, P. Karras, C. Raïssi, and K.-L. Tan, “ρ-uncertainty: inference-proof transaction anonymization,” *Proceedings of the VLDB Endowment*, vol. 3, no. 1–2, pp. 1033–1044, Sep. 2010.

[7] arx-deidentifier and F. Prasser, “An overview of methods for data anonymization,” https://de.slideshare.net/arx-deidentifier/prasser-methods, 2015.

[8] M. Gentili, S. Hajian, and C. Castillo, “A Case Study of Anonymization of Medical Surveys,” in *Proceedings of the 2017 International Conference on Digital Health - DH ’17*, London, United Kingdom, 2017, pp. 77–81.

[9] K. El Emam and B. Malin, “Concepts And methods for de-identifying clinical trial data,” *Paper commissioned by the Committee on Strategies for Responsible Sharing of Clinical Trial Data*, 2014.

[10] N. Johnson, J. P. Near, and D. Song, “Towards Practical Differential Privacy for SQL Queries,” p. 14, 2018.

[11] N. V. Dijkhuizen and J. V. D. Ham, “A Survey of Network Traffic Anonymisation Techniques and Implementations,” *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–27, May 2018.

[12] L. Sweeney, “Guaranteeing anonymity when sharing medical data, the Datafly System.,” *Proc AMIA Annu Fall Symp*, pp. 51–55, 1997.

[13] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Incognito: efficient full-domain K-anonymity,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD ’05*, Baltimore, Maryland, 2005, p. 49.

[14] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Mondrian Multidimensional K-Anonymity,” in *22nd International Conference on Data Engineering (ICDE’06)*, Atlanta, GA, USA, 2006, pp. 25–25.

[15] R. J. Bayardo and R. Agrawal, “Data Privacy through Optimal k-Anonymization,” in *21st International Conference on Data Engineering (ICDE’05)*, Tokyo, Japan, 2005, pp. 217–228.